

(Erweiterte Inbetriebnahmephase) Whitepaper zur Anbindung an VIDIS für Service Provider

Version: 60
Exportiert am 13-08-2024



Inhaltsverzeichnis

Inhaltsverzeichnis	2
1. Informationen für Service Provider (SP).....	3
2. Pilotsystem	3
2.1. Anbindung eines OpenID Connect Client	3
2.1.1. FWU/VIDIS an SP zusammengefasst.....	3
2.1.2. SP an FWU/VIDIS zusammengefasst.....	4
2.2. Beschreibung OIDC-Claims	4
2.3. Forschungs ID.....	6
2.4. Gruppen und Klassendaten über SchulConneX Schnittstelle	6
3. Weitere technische Voraussetzungen	8
3.1. Automatische Registrierung bei initialer Anmeldung	8
3.2. Logout.....	8
3.3. Vorauswahl des IDP (vidis_idp_hint).....	9
4. FAQ.....	9
4.1. Userinfo überprüfen	9
5. Integration VIDIS-Login	10
5.1. Was ist der VIDIS-Login?.....	10
5.2. Voraussetzungen für die Integration des VIDIS-Login:.....	10
5.3. Integration des VIDIS-Login Buttons:	10
6. Fragen & Anregungen	12

1. Informationen für Service Provider (SP)

Um die geordnete Anbindung eines "Service Providers" an das VIDIS-System zu ermöglichen, müssen entsprechende Metadaten gemäß <https://openid.net/specs/openid-connect-discovery-1.0.html> ausgetauscht werden.

In diesem Dokument wird ausschließlich auf den Fall einer Anbindung von Angeboten (Webseiten und Apps) als OpenID Connect Client eingegangen.

Es werden in späteren Projektphasen noch weitere SSO-Technologien für den Einsatz evaluiert. Es ist jedoch unwahrscheinlich, dass der produktive VIDIS-Dienst später viele weitere SSO-Technologien unterstützen wird.

2. Pilotsystem

Ein Testsystem ist derzeit in Betrieb. Alle Tests, insbesondere die Test-Anbindungen, werden derzeit an diesem VIDIS-System durchgeführt. Dieses Testsystem dient dauerhaft als Integrations- und Testsystem für die VIDIS-Infrastruktur und eine Anbindung ist Voraussetzung für die Anbindung an das Pilot- und Produktivsystem.

2.1. Anbindung eines OpenID Connect Client

Für die Anbindung eines OpenID Connect Clients müssen folgende Parameter ausgetauscht und sowohl im VIDIS-System als auch beim anzubindenden OpenID Connect Client konfiguriert werden.

2.1.1. FWU/VIDIS an SP zusammengefasst

Testsystem

- **"ClientID"**: der Identifikator des anzubindenden OpenID Connect Clients
- **"Client Secret"**: ein zwischen VIDIS und dem OpenID Connect Client geteiltes Geheimnis
- **"Authorize Endpoint"**: <https://aai-test.vidis.schule/auth/realms/vidis/protocol/openid-connect/auth>
- **"Access Token Endpoint"**: <https://aai-test.vidis.schule/auth/realms/vidis/protocol/openid-connect/token>
- **"end_session_endpoint"** <https://aai-test.vidis.schule/auth/realms/vidis/protocol/openid-connect/logout>

Die maschinenlesbare OpenID Connect Konfiguration befindet sich zusammengefasst unter: <https://aai-test.vidis.schule/auth/realms/vidis/.well-known/openid-configuration>

Pilot- und Produktivsystem

- **"ClientID"**: der Identifikator des anzubindenden OpenID Connect Clients
- **"Client Secret"**: ein zwischen VIDIS und dem OpenID Connect Client geteiltes Geheimnis
- **"Authorize Endpoint"**: <https://aai.vidis.schule/auth/realms/vidis/protocol/openid-connect/auth>
- **"Access Token Endpoint"**: <https://aai.vidis.schule/auth/realms/vidis/protocol/openid-connect/token>

- **"end_session_endpoint"** <https://aai.vidis.schule/auth/realms/vidis/protocol/openid-connect/logout>

Die maschinenlesbare OpenID Connect Konfiguration befindet sich zusammengefasst unter: <https://aai.vidis.schule/auth/realms/vidis/.well-known/openid-configuration>

2.1.2. SP an FWU/VIDIS zusammengefasst

Notwendig:

- **"Valid Redirect URIs"**: zulässige Redirect URIs
- **"BaseURL"**: zum Testen der Verbindung von FWU-Seite (Das ist der Einstiegspunkt bzw. das Login Formular)
- **"Deeplink" ins Angebot**: Wird benötigt, um von Landesportalen mit Session direkt in Ihr Angebot zu springen ohne erneute Anmeldung. Dafür muss ein Parameter angehängt werden → "kc_idp_hint" (siehe unten: Weitere technische Voraussetzungen)

Optional:

- **"Backchannel-Logout-URL"**: zulässige Logout-URL
- Social-Media-Vorschau bild für Deeplink: Hilfreich bei Verlinkung bzw. Anzeige in Lern-Management-Systemen, Mediatheken und Portalen

2.2. Beschreibung OIDC-Claims

OIDC-Name	Beschreibung	Typ	Auslieferung	Verfügbarkeit	Beispiel	Multivalu e
sub	Eineindeutige ID (Sub, ggf. Pseudonym, ID-Token)	String	Token + userinfo-Endpoint	Pflichtfeld	e8c4cc50-d2e1-4de3-90c7-2262494f6121	Nein
akronym	Abgekürzter Personenbezeichner zu Anzeigezwecken (die ersten beiden Anfangsbuchstaben von Vor- und Nachname)	String	userinfo-Endpoint	Optional	HaWu	Nein
schulkennung	Schul-Identifizier der Stammschule (eineindeutige länderübergreifende Kennung, bestehend aus Landeskennung + landespezifischer Schul-ID)	Array	Token + userinfo-Endpoint	Pflichtfeld	DE-BY-12345	Ja

OIDC-Name	Beschreibung	Typ	Auslieferung	Verfügbarkeit	Beispiel	Multivalu e
bundesland	Bundesland	String	Token + userinfo- Endpunkt	Pflichtfeld	DE-BY	Nein
heimatorganisation	Identity-Provider	String	Token + userinfo- Endpunkt	Optional	DE-NI-SANIS, DE-SN- EVLKS, DE-SN- Schullogin	Nein
rolle	Rolle in der Stammschule für den Service Provider	String (LEHR / LERN / LEIT)	Token + userinfo- Endpunkt	Pflichtfeld	LERN	Nein
vorname	Vorname (Wird nur für Lehrkräfte übermittelt)	String	Token + userinfo- Endpunkt	Optional	Max	Nein
nachname	Nachname (Wird nur für Lehrkräfte übermittelt)	String	Token + userinfo- Endpunkt	Optional	Mustermann	Nein
email	Email Adresse aus dem IDM oder pseudonymisiert von VIDIS	String	Token + userinfo- Endpunkt	Optional	e8c4cc50-d2e1-4de3- 90c7- 2262494f6121@vidis.sch ule	Nein
lizenzen	Lizenz Informationen, separiert durch Hash	Array	userinfo- Endpunkt	Abhängigkeit zu LC (perspektivisch Q3 2025)	-	Ja
SchulConneX	Unterstützung verschiedener Rollen, mehrerer Schulen, Gruppen und Klassendaten im Personenkontext	Object	Token + userinfo- Endpunkt	Optional wenn vom IDP unterstützt (Aktuell nur NI)	Siehe Abschnitt - <i>Gruppen und Klassendaten über SchulConneX Schnittstelle</i>	-
Forschungs ID	Eineindeutige ID Angebotsübergreifend im Bundesland	String	Token + userinfo- Endpunkt	Optional	z5c4cc50-d2e1-4de3- 90c7-2262494f6846	Nein
OIDC Standard Claims						

OIDC-Name	Beschreibung	Typ	Auslieferung	Verfügbarkeit	Beispiel	Multivalu e
Scope: -	Nicht implementiert: name, family_name, given_name, middle_name, nickname, picture, updated_at, email, email_verified	-	-	-	-	-

```

{
  "exp": 1668155380,
  "iat": 1668155080,
  "auth_time": 0,
  "jti": "0a081ce3-5e15-4d85-9919-4ec95c2051ea",
  "iss": "https://aai-test.vidis.schule/auth/realms/vidis",
  "aud": "client-alias",
  "sub": "f3738cf7-a646-4bd7-af61-4a4c9e8151ef",
  "typ": "ID",
  "azp": "client-alias",
  "session_state": "b6b0b7f9-daf6-4377-a2e6-965f3356b3ba",
  "acr": "1",
  "sid": "b6b0b7f9-daf6-4377-a2e6-965f3356b3ba",
  "rolle": "LEHR",
  "schulkennung": [
    "DE-LAND-12345"
  ],
  "bundesland": "DE-LAND",
  "heimatorganisation": "DE-LAND-Schulportal"
}

```

Beispiel JSON - Standard Attribute von VIDIS

2.3. Forschungs ID

Ein Forschungsverbund in Brandenburg (bestehend aus der Technischen Universität Dortmund, dem Deutschen Zentrum für Lehrkräftebildung Mathematik und dem DIPF | Leibniz-Institut für Bildungsforschung und Bildungsinformation) nutzt im Projekt "Abako" personalisierte Bildungsmedien über VIDIS und die Plattform "alea.schule". Dabei werden personenbezogene Daten mit VIDIS-Pseudonymen gesammelt. Zusätzlich zur VIDIS-Pseudonym-ID erhält der Forschungsverbund vom Land Brandenburg die Genehmigung, eine Angebotsübergreifende Forschungs-Scope-ID zu verwenden. Diese Forschungs-ID wird einem kurzen Probandencode zugeordnet, den Lehrkräfte gut merken können. Die Zuordnung erfolgt entweder durch einen Hashing-Algorithmus oder eine Konkordanzliste, die im Forschungsverbund mit einer Löschfrist von 10 Jahren gespeichert wird. In weiteren Befragungen im Rahmen des "Abako"-Projekts geben Lehrkräfte zusätzlich zu ihren Antworten diesen "Probandencode" an. : Originaltext gekürzt und verallgemeinert.

2.4. Gruppen und Klassendaten über SchulConneX Schnittstelle

Wir können auch Gruppen und Klassendaten weitergeben wenn diese im Landesportal zur Verfügung stehen. Dies ist wie folgt geplant, und wird aktuell getestet, ist jedoch noch kein Bestandteil unseres aktuellen

Standards. Hier finden Sie den Releaskandidat für die Version 1.4:

https://github.com/SchulConneX/v1/blob/main/SchulConneX_Schnittstellenspezifikation_v1.4_Veroeffentlichungskandidat.pdf

Attribute Key	Type	Description
person.kontext.<number>.id	String	ID des Kontexts
person.kontext.<number>.referrer	String	ID der Gruppe im Quellsystem. Wird vom Quellsystem vergeben und muss im Quellsystem eindeutig sein.
person.kontext.<number>.org.id	String	ID der Organisation
person.kontext.<number>.org.kennung	String	Kennung der Schule (Schulidentifikator)
person.kontext.<number>.org.vidis_schulidentifikator	String	Vidis Schulidentifikator
person.kontext.<number>.org.name	String	Name der Schule
person.kontext.<number>.org.typ	String	Typ der Organisation
person.kontext.<number>.rolle	String	Roller der Person
person.kontext.<number>.status	String	Status
person.kontext.<number>.gruppen	JSON	Gruppen und Zugehörigkeit
person.kontext.<number>.loeschung	JSON	Zeitpunkt der Löschung

```

"gruppen": [
  {
    "gruppe": {
      "id": "ab34d607-b950-41a5-b69d-80b8812c224a",
      "mandant":
"02feb60dc3f691af4a4bf92410fac8292bb8e7d6adebb70b2a65d3c35d825d8a",
      "orgid":
"02feb60dc3f691af4a4bf92410fac8292bb8e7d6adebb70b2a65d3c35d825d8a",
      "referrer": "fe4e50cb-c148-4156-8c2f-dc5260b267cf",
      "bezeichnung": "Englisch, 2. Klasse",
      "thema": "Thema",
      "beschreibung": "Beschreibung der Gruppe",
      "typ": "SONSTIG",
      "bereich": "WAHL",
      "optionen": [
        "01",
        "02"
      ],
      "differenzierung": "G",
      "bildungsziele": [
        "GS"
      ],
      "jahrgangsstufen": [
        "JS_02"
      ],
      "faecher": [
        {
          "code": "EN"
        }
      ],
    },
  ],
]

```

Codeblock 1 Beispiel JSON

3. Weitere technische Voraussetzungen

3.1. Automatische Registrierung bei initialer Anmeldung

Voraussetzung für Service Provider ist, dass Nutzerinnen und Nutzer, die sich an dem digitalen Bildungsangebot erstmalig anmelden, bei der Anmeldung automatisch registriert werden.

3.2. Logout

Beim Logout im Bildungsangebot muss der VIDIS-Logoutendpoint aufgerufen werden damit VIDIS die Usersession beenden und den User löschen kann. Wird dies nicht gemacht, kann VIDIS den User nicht automatisch löschen.

Endpoint Pilotsystem: <https://aai.vidis.schule/auth/realms/vidis/protocol/openid-connect/logout>

Endpoint Testsystem: <https://aai-test.vidis.schule/auth/realms/vidis/protocol/openid-connect/logout>

Die Bestätigung über den Logout kann nur geskippt werden wenn `post_logout_redirect_uri` und `id_token_hint` Paramter in der URL gesetzt sind.

post_logout_redirect_uri ist die URI auf die nach dem Logout weitergeleitet werden soll.

Der ID_TOKEN_HINT benötigt den ID-Token als Inhalt.

Dies ist verpflichtend für Serviceprovider.

3.3. Vorauswahl des IDP (vidis_idp_hint)

Voraussetzung für Service Provider ist, die automatische Weitergabe des Parameter vidis_idp_hint (z.B. "?vidis_idp_hint=Landessystem") während der Anmeldung. Für einige Anwendungsfälle (z.B. Direktaufruf aus Landessystemen heraus) ist die Vorauswahl eines Landes-IdP wichtig. Der VIDIS-Dienst unterstützt diese Vorauswahl durch Übergabe des Parameter vidis_idp_hint. Dieser Parameter muss also bei aufrufen (z.B. bei Authorization Requests) unverändert weitergegeben werden und darf nicht herausgefiltert werden.

4. FAQ

4.1. Userinfo überprüfen

Über den endpoint <https://aai.vidis.schule/auth/realms/vidis/protocol/openid-connect/userinfo> können Sie sich die Übermittelten Userdaten einsehen.

Das funktioniert wie folgt.

Schritt 1: Zuerst muss ein Token generiert werden.

Dieser Token (access_token) fungiert als "Bearer Token" der zum Abruf der User Info genutzt werden kann und zu Debugging zwecken, der Redirect Flow wird bei der finalen Integration verwendet.

```
curl --location --request POST
'https://aai.vidis.schule/auth/realms/vidis/protocol/openid-connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'username=XXXXX' \
--data-urlencode 'password=XXXXX' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'client_id=account'
```

Schritt 2: Jetzt lassen sich die Userdaten mit dem Token abrufen

```
curl --location --request GET
'https://aai.vidis.schule/auth/realms/vidis/protocol/openid-connect/userinfo' \
--header 'Authorization: Bearer <TOKEN>
```

oder alternative zu Schritt 2 wäre es den JWT Token zu dekodieren, um die Nutzerdaten auslesen zu können. Hierzu kann beispielsweise auch ein entsprechendes Online Tool verwendet werden (z. B. über <https://devtoolzone.com/decoder/jwt>, <https://jwt.io/> etc.), sofern es sich nicht um reale Nutzerdaten handelt.

Das Ergebnis ist ein JSON mit den Userdaten:

```
{  
  "sub": "39e0063a-8377-4a1c-bd15-ea16f65a5a15",  
}
```

5. Integration VIDIS-Login

5.1. Was ist der VIDIS-Login?

Das Ziel des VIDIS-Logins ist es, jedem Schüler und Lehrer ein einziges Konto zur Verfügung zu stellen, mit dem man sich überall einloggen kann.

Dazu wird eine bestehende Schul-Login als IDP verwendet. Das bedeutet, dass der VIDIS-Login den Benutzer in das IDP-System einloggt, das die Schule verwendet und zu Ihnen zurückkehrt.

Der Benutzer muss also die IDP auswählen, die er verwenden kann, was beispielsweise auch über den VIDIS-Button geschehen kann.

Die Funktionalität kann hier getestet werden: <https://tp.fwu.saas-dev.aws.intension.eu/?version=latest>

5.2. Voraussetzungen für die Integration des VIDIS-Login:

- Unterstützung OpenID-Connect

5.3. Integration des VIDIS-Login Buttons:

Allgemeine Vorgehensweise:

1. Hinzufügen eines CDN-Links
2. Bauen Sie die Webkomponente des Vidis Login-Buttons ein
3. Geben Sie dem Button Ihren Login-Link
4. Konfiguration des Vidis Login-Buttons
 - a. Größe (size)
 - b. Cookie (cookie)

1. Hinzufügen eines CDN-Links

Damit der Vidis Login Button funktioniert, müssen Sie einen CDN-Link in Ihre Website einbinden:

```
1 <script src="https://repo.vidis.schule/repository/vidis-  
  cdn/latest/vidisLogin.umd.js"></script>
```

Der Link, den Sie normalerweise verwenden würden, lautet: <https://repo.vidis.schule/repository/vidis-cdn/latest/vidisLogin.umd.js>

Wenn Sie eine bestimmte Version bevorzugen: <https://repo.vidis.schule/repository/vidis-cdn/{version}/vidisLogin.umd.js>

Zum Beispiel: <https://repo.vidis.schule/repository/vidis-cdn/0.11.0/vidisLogin.umd.js>

2. Bauen Sie die Webkomponente des Vidis Login-Buttons ein

Wenn Sie sich entschieden haben, wo der Vidis-Login-Button platziert werden soll, können Sie ihn wie folgt hinzufügen:

```
1 <vidis-login loginurl=""></vidis-login>
```

3. Geben Sie den Login-Link für den Button an

WICHTIG: Sie müssen die Login-URL Ihres Systems angeben, sonst kann der Button nicht funktionieren.

WICHTIG: Die Login-URL muss mit "https://" beginnen, um erkannt zu werden.

Der Benutzer wird umgeleitet, indem die angegebene Login-URL mit diesem Abfrageparameter ergänzt wird: kc_idp_hint

4. Konfiguration des Vidis Login Buttons

Sie können den Vidis Login Button mit den folgenden Attributen anpassen:

- Größe: Bestimmt die Größe des Buttons.
 - Werte:
 - "L": Groß, zeigt die Schaltfläche in einer großen Version an. Dies ist auch die Standardeinstellung.
 - "M": Mittel, zeigt die Schaltfläche in einer mittleren Version an.
 - "S": Klein, zeigt die Schaltfläche in einer kleinen Version an.
 - Cookie: Aktiviert oder deaktiviert die Speicherung der letzten Auswahl des Benutzers in einem Cookie.
 - Es wird empfohlen, diese Option zunächst auf "false" zu setzen und erst dann zu aktivieren, wenn der Benutzer den Cookies auf Ihrer Website zugestimmt hat, um rechtliche Probleme zu vermeiden.

Ein vollständiges Beispiel:

VIDIS-Login Example

```
1 <script src="https://repo.vidis.schule/repository/vidis-  
2 cdn/1.0.1/vidisLogin.umd.js"></script>  
3 ...  
  <vidis-login loginurl="https://www.domain.de/path-to-  
  auth/" size="L" cookie="true"></vidis-login>
```

Kompatibilität:

Der Vidis Login Button ist als Webkomponente erstellt und sollte daher in jeder html-basierten Umgebung funktionieren, insbesondere in jedem SPI-Framework wie Vue, Angular und React.

Technisch gesehen, wenn man weiß, wie man eine Webkomponente in andere Apps (wie Android oder IOS) integriert, sollte der Button auch out of the box funktionieren, ist aber noch nicht dafür getestet.

6. Fragen & Anregungen

Für Fragen und Anregungen melden Sie sich gerne jederzeit unter vidis@fwu.de